MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No.927 January 1988

Stereo and Eye Movement

Davi Geiger and Alan Yuille

Abstract: We describe a method to solve the stereo correspondence using controlled eye (or camera) movements. These eye-movements essentially supply additional image-frames which can be used to constrain the stereo matching. Because the eye-movements are small, traditional methods of stereo with multiple frame will not work. We develop an alternative approach using a systematic analysis to define a probability distribution for the errors. Our matching strategy then matches the most probable points first, thereby reducing the ambiguity for the remaing matches. We demonstrate this algorithm with several examples.

This report describes research done within the Artificial Intelligence Laboratory. Support for the A.I. Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research (ONR) contract N00014–85–K–0124.

1 Introduction

This work examines the use of eye (or camera) movements to help solve the stereo correspondence problem.

1.1 Stereopsis assumptions and violations

A key problem of stereo vision is to determine the correspondence between features in the two eyes. These features could be image intensity values, edges or other primitives. This correspondence is, in general, underdetermined and heuristics are needed to solve it. These heuristics are derived from expectations about the world. For example, most surfaces in the world are smooth and opaque and this gives rise to the ordering constraint; points usually lie in the same order on corresponding epipolar lines in the two images. Other common heuristics used for matching are the coarse to fine strategy (there are fewer features at larger scales and therefore less ambiguity) and figural continuity (edges tend to have the same depth). Assumptions of these types are necessary for stereo correspondence, but do not always hold. An example where they fail is the double nail illusion (Krol and Van de Grind 1982)[3], shown in figure 1. Assume two points in space are fixed with the same coordinates x and z but slightly different depth y. They project into two points in each eye. The correct match is given when the leftmost point in the right eye matches to the rightmost point in the left eye and vice versa for the other match. The ordering constraint gives the wrong matches and indeed psychophysical experiments show that humans also make this mistake. Most stereo algorithms make use of the ordering constraint, either implicitly or explicitly, and would fail on this example.

1.2 Stereo with eye movement

Eye movements are an alternative method which could be used to solve the correspondence problem. By rotating the eyes to alter the direction of fixation we introduce extra views of the same object (we use the same model of eye-rotation as Longuet-Higgins 1982 [9], see figure 4). This corresponds to having several views of the object and for machine vision is similar to doing stereo with three or more cameras [13]. The correspondence problem is whether point A in the right image matches point B in the left image. In

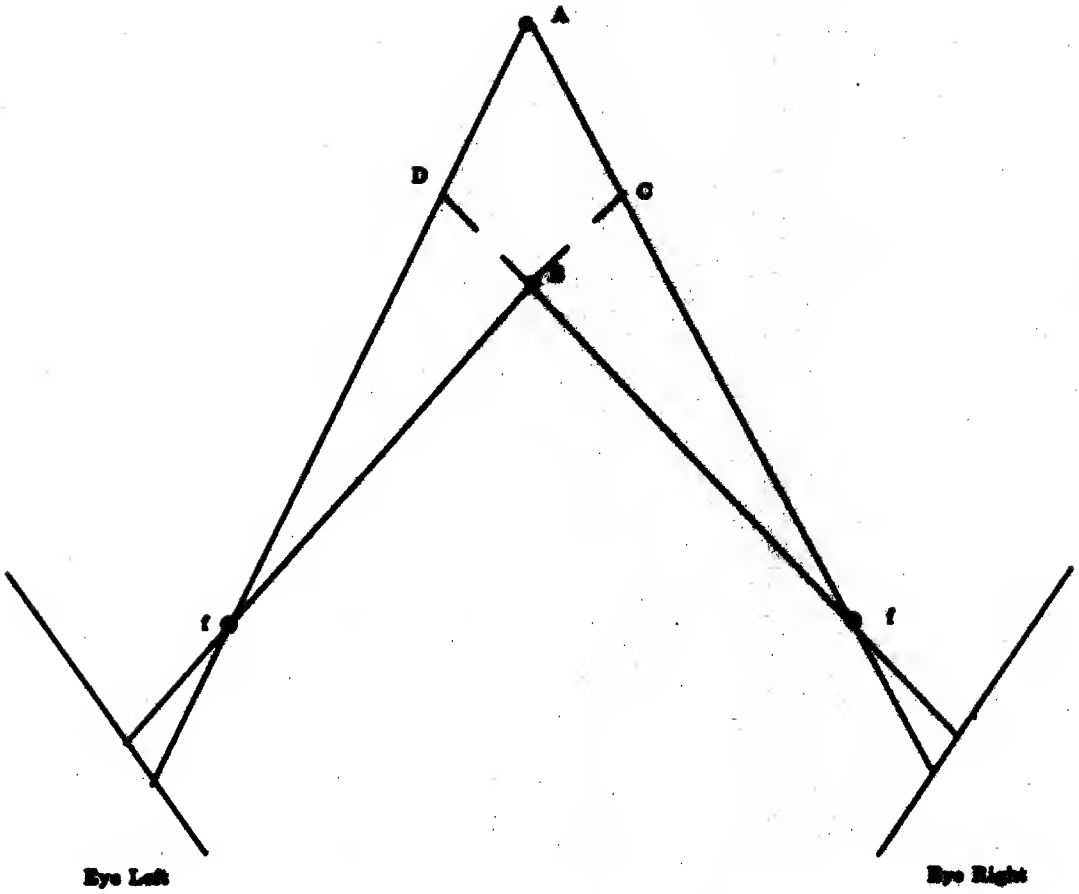

Figure 1: *The Double Nail illusion. The ordering constraint is violated. The physical points A and B generate the illusion of points C and D.*

each extra view of the scene there is an epipolar line associated to point A and a $different$ one associated to point B. The multiple frame idea is: if there is in the intersection point of both epipolar lines (associated to A and B) an image point, than A matches B. Otherwise it does not. The idea is illustrated in figure 2. Unfortunately, due to the finite size of the image lattice and the limited size of the rotation angles, this method will not work well for eye movement. The additional frames are too close to the first two and many false matches could occur. However we can adapt this method as the basis of a stereo test. We define a *stereo test*, if there is a match between two points then they correspond to a unique point in 3D-space and the projections of this point must appear in the additional frames (see figure 2. If no points are seen in the additional frames at the predicted positions then the hypothetical match fails the test. Unless the images are sparse there will be too many points passing the stereo test.

1.3 A strategy for using eye movement to help solve the correspondence problem

Because eye movements are small it is generally believed that they may only yield very weak information. We now introduce an algorithm where they can be used to help solve the correspondence problem. The algorithm is precisely described in chapter 3 and 4.
We limit ourselves to objects with distinguished features, such as dots. In the final section we show how the strategy proposed below can be extended to more realistic scenes.
The proposed strategy falls into two parts, see figure 3. First we track (and match) features in the left and right eyes separately. Each eye gives a rough estimate for the 3-D position of the point and the error range for this position. We now use these estimates as the basis for the stereo match. Second we define a *rotation depth test*, which accepts a possible match if the estimated positions and the error range are compatible, and also a *ratio test* (described later). The strictness of these tests depends on a set of *control parameters*. Initially these parameters are set to make the tests very difficult. The program now hypothesizes matches between points in the left and right eyes. If the tests are all passed these matches are accepted and the points are not considered further. Then the algorithm changes the control parameters

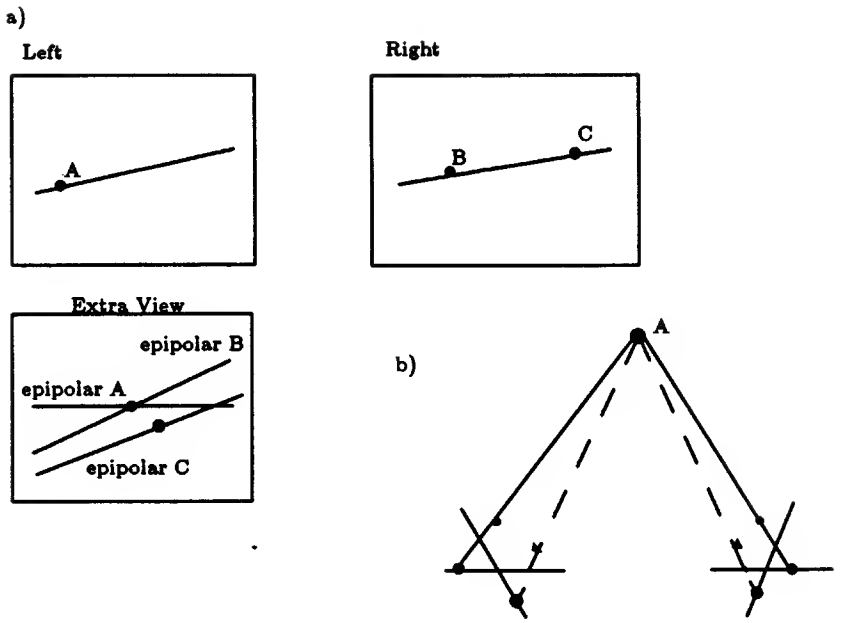

Figure 2: *a) Point A in the left image matches point B in the right image (and not point C) since the epipolar lines associated with them in the extra view intersect in an image point. b) A point projected in four frames, using the Longuet-Higgins model of eye-rotation.*

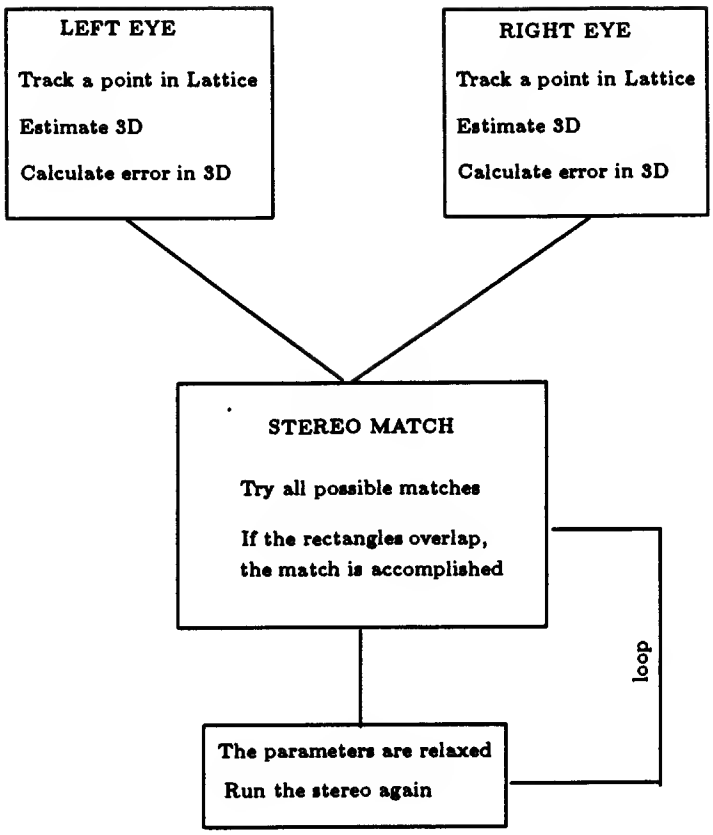

Figure 3: *Flow chart of the strategy for the matching process.*

to reduce the strictness of the test and matches are again hypothesized and tested. This procedure is repeated until the control parameters reach a final value. Points which have not been matched by the algorithm, perhaps because they occur at occluding boundaries and are only visible to one eye, are assigned the 3-d position estimated by the rotation of the eyes. A "zooming" feature can be added to the basic algorithm: if a certain region of the image contains a large number of points the eyes can zoom in to this region by changing the focal length, thereby increasing the resolution.

Error estimation is an important aspect of this algorithm. For every point in each eye we use eye-rotation to estimate the depth of the 3-D point corre-

sponding to it. Errors in this estimate arise from the finite size of the lattice. We derive a probability distribution for such errors. Thus for each point in the eye we have a probability distribution for its position in space. The more the overlap of the probability distributions of a pair of points (one in each eye) the more likely they are to correspond. Our strategy essentially matches the points with overlaps above a certain threshold, removes these points thereby reducing the ambiguity for the remaining matches, lowers the threshold and repeats the process.

We have implemented our algorithm on a Symbolics LISP machine and tested it on a variety of synthetic images. The algorithm is inherently parallelizable. We plan to implement it on the Connection machine and attach it to the MIT head-eye system [2].

We have chosen examples which would be difficult for conventional stereo algorithms because of the lack of a smooth surface, the occurance occluding boundaries and the violation of the ordering constraint (although models based on the disparity gradient limit [16] [6], work on some of these stimuli). We start with a cube figure with features at regular intervals along the boundaries. The cube is transparent, so the adjacent features do not lie on the same surface and the ordering constraint is sometimes violated. We test the algorithm on the double nail illusion and show that it only makes mistakes when the nails are extremely close together. A third example is the occluding boundary of a circular figure. Because the surface turns smoothly away from the viewer the two eyes will see the boundary at different points. We show how our algorithm can check to see if this occurs. Finally we consider a transparent random dot stereogram.

In the next section we discuss eye movements. The following sections describe the mathematics of our eye-system, the error analysis and the description of the algorithm. We then illustrate the algorithms on the examples described above, describe extensions of this work to real images and to stereo with more general motion.

2 Eye movements

2.1 When and Why eye movements is important

It was originally thought that the role of eye movement was merely to retain the object of perception in the visual field and to change the points of fixation, but the subject is far more complex. Under natural conditions the human eye never ceases moving (these movements are called saccades) and if an object is artificially kept strictly stationary relative to the retina for about 3 seconds or longer it fades. It has been shown [19] that slight movement of the retinal image over the retina (as caused by eye movement) is necessary for optimal perception. The result of this movement of the retinal image is to cause the light stimulating the receptors to be constantly changing. Electrophysiological studies [19] have shown that in many animals electric impulses appear in the optic nerve specifically in response to a change in the light acting on the retina. More precisely, it is known that most ganglion cells are transient. These movements are small but to quote Yarbus "when understood, the role of eye movements and the principles governing these movements may help to solve many purely practical problems." Poggio and Poggio 1984 [15] state that "Given the obvious importance of eye movements in stereopsis, it is surprising that so little is known about the role of vergence." We argue that eye movements can be helpful for stereopsis. We will be considering controlled eye rotation rather than the apparently random fluctuations of saccades.

From a computational perspective the critical problem in stereopsis is the matching process between the two eyes [11]. We argue that eye movement can help provide the necessary constraints for the matching process. To do this we need precise estimates of angles of rotation of the eyes. Humans have very good accuracy at angle estimation when they rotate their eyes [1]. Longuet-Higgins [9] (see also Mayhew [12]) has shown that information to estimate these angles is directly available. However it is not clear whether humans use this information.

In practice eye movement does not seem to provide as much help for stereopsis as we have suggested. A possible reason comes from the small size of the fovea. If the eyes rotate by a large angle the fovea will view different scenes.

[1] Conversation with David Robinson

Therefore the information available will not be different views of the same scene but instead will be different views of **different** scenes. By including the whole macula lutea we increase the range of allowed rotation angles, but at the expense of resolution. Our work may reflect an overestimation of the information retrieved from eye movement, however the potential help from eye movement should at least appear in some level of the stereopsis process. For example it could be used for registering the image in each eye.
We have to point out that stereo is possible when eye movements are not allowed. However, more systematic work to compare both situations (eye-movement and no-movement) is required to decide on how does the brain do stereo. In any case we show here that eye (or rather camera) movement can be an important tool for machine vision. Eye movements can also give an estimate of depth if part of an object is occluded and only visible to one eye. Humans are less good at doing stereopsis on transparent objects, and this is a source of immense error for most stereo algorithms. The two basic reasons for those errors are due to the assumptions of ordering constraint and of smooth surfaces between edges. Both assumptions are used as constraints for the matching process. However they are badly violated for transparent objects. We show that eye movement can provide enough information to constraint the stereo matching and those two assumptions do not have to be used strongly in our algorithm. Thus we expect that our algorithm will work well also for transparent objects. Another example is the Double Nail illusion where eye movements can be very helpful. This illusion is illustrated in figure 1. Since the ordering constraint is violated, an algorithm based on ordering constraint gives the wrong matches. The use of eye movements, with sufficient rotation, gives a clue for the depth of each point. This clue is sufficient to give the correct match. The human visual system often gets the wrong matches in this case. Our algorithm, however, correctly solve the double nail illusion.

3 The Basic Geometry

The geometrical model of the head is the Longuet-Higgins model illustrated in figure 4 where each eye is allowed to rotate around the z axis. The center of the right-eye and left-eye are at

$$\vec{O}_r = (l, 0, 0) \qquad and \qquad \vec{O}_l = (-l, 0, 0).$$

So the distance from the center of the head to each eye is l. We use perspective projection with the right and left foci at

$\vec{F}_r = (l + fsin\Phi, fcos\Phi, 0) \qquad \vec{F}_l = (-l + fsin\Psi, fcos\Psi, 0)$

where Ψ and Φ are the angles of rotation around the z axis.

3.1 Projection of a point in the screen

Given a point $\vec{X} = (x, y, z)$ we define P_x to be the projection of this point onto the screen (for the right eye). The coordinates of P_x in terms of the coordinate system of the screen are given by

$$x_R = -f\frac{(x-l)cos\Phi - ysin\Phi}{(x-l)sin\Phi + ycos\Phi - f}$$

$$z_R = -f\frac{z}{(x-l)sin\Phi + ycos\Phi - f}. \tag{3.11}$$

Similarly for the left eye we have

$$x_L = -f\frac{(x+l)cos\Psi - ysin\Psi}{(x+l)sin\Psi + ycos\Psi - f}$$

$$z_L = -f\frac{z}{(x+l)sin\Psi + ycos\Psi - f} \tag{3.12}$$

3.2 Epipolar lines

Epipolar lines between eyes at different rotations

A projected point P_{Ox1} in the right eye at angle Φ_1 corresponds to the projection of all the points generated by a line $\vec{L}$. If we rotate the eye to an angle Φ_2 , this line will not be projected into a single point any longer but will be projected into a line. This line is the epipolar line associated to the point P_{Ox1}. From 3.11 with $x_l = (x - l)$ we obtain

$$z_{R1}[-x_{R2}(1 - cos(\Phi_2 - \Phi_1)) - fsin(\Phi_2 - \Phi_1)] =$$

$$= z_{R2}[x_{R1}(1 - cos(\Phi_2 - \Phi_1)) - fsin(\Phi_2 - \Phi_1)] \tag{3.21}$$

where the sub-index 1(2) refers to the first (second) frame. Equation (3.21) is linear in the variables x_{R2} and z_{R2} since $\Phi_1, \Phi_2, x_{R1}, z_{R1}$ are given. This

linear equation defines the epipolar lines in the rotated frame at angle Φ_2. For the left-eye we conclude by analogy that

$$z_{L1}[-x_{L2}(1 - cos(\Psi_2 - \Psi_1)) - f sin(\Psi_2 - \Psi_1)] =$$

$$= z_{L2}[x_{L1}(1 - cos(\Psi_2 - \Psi_1)) - f sin(\Psi_2 - \Psi_1)]. \quad (3.22)$$

Epipolar lines between left and right eyes
The epipolar line between left and right eyes is defined in a similar way as for eyes with different rotation angles. More precisely, given a projected point P_{Oxr}, in the eye-right at angle Φ_1, it corresponds to the projection of all the set of points generated by the line $\vec{L}$. This line when projected in the left eye generates the epipolar line represented by $\vec{P}_{Oxl}$. So from (3.11), (3.12) we conclude

$$z_R[2lx_L sin\Psi + fx_L(cos(\Psi - \Phi) - 1) - f^2 sin(\Psi - \Phi) + 2flcos\Psi] =$$

$$= z_R[2lx_R sin\Phi + fx_R(1 - cos(\Psi - \Phi)) - f^2 sin(\Psi - \Phi) + 2flcos\Phi]. \quad (3.23)$$

Given the projected point in the right image, more precisely given Φ, Ψ, x_R and z_R, (3.23) becomes a linear equation in x_L and z_L. The solution of (3.23) gives the epipolar lines in the left eye.

4 The eye system and the artificial eye system

In this section we discuss the relationship between our artificial model and the human eye. We compare the resolutions of the two systems. We show how our results scale with depth, angle of rotation and lattice size. These results suggest that our algorithm could be useful for objects at depths of the order of up to several meters.

Figure 4 illustrates the eye system. Light enters through the cornea, is refracted according to Snell's law into the eyes. It then is focused by the lens, and is "seen" by the macula lutea, which includes the fovea. The diameter of the eye is about *24 mm* in all directions. The eye rotates about a point on the central axis at a distance of *14.5 mm* from the cornea. This model

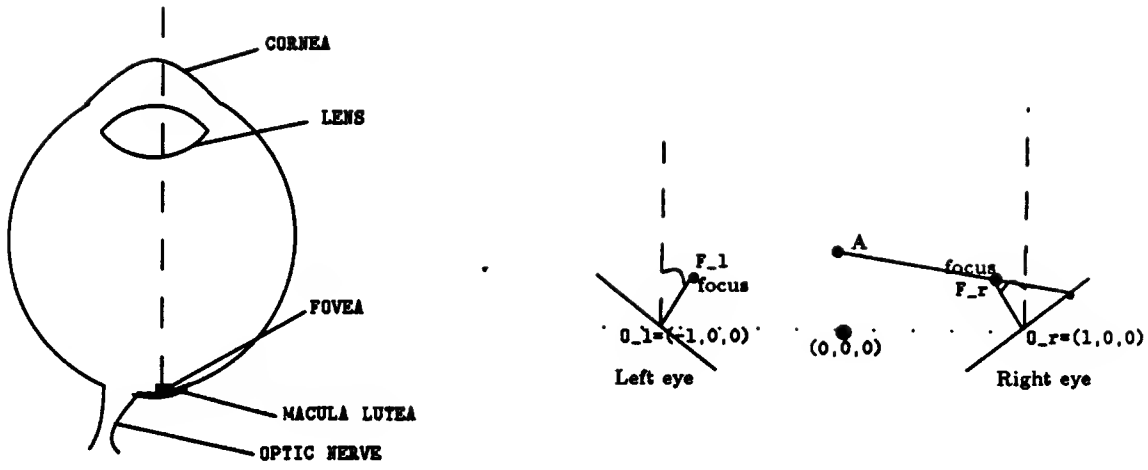

Figure 4: *Model of the eye system and Longuet-Higgins model.*

is equivalent of having the eye to rotate about a virtual fovea (located at *14.5 mm*) and with a virtual focus reduced to approximate *10 mm*. Figure 5 illustrate the equivalence between the eye and the virtual eye. What we must avoid, however, is having the eye rotate about its focus. It is easy to see that for this case no additional information arises when the eye is rotated.

The resolution of a vision system, humans eye or cameras, possesses two parameters that are interconnected; the size of the lattice and the focal length. The size of the lattice in the human eye is fixed (it has some spatially variation) and is given by the density of 1 cone per 0.0014 mm in the region of the fovea. The value of the focus can vary, a typical value is about *17 mm*. This implies that each cone is responsible for *25*" of an arc. Those numbers are not so easy to obtain since we have to take into account Snell's law for the incident light from the air to the liquid inside the eyes. Note that the hyperacuity of human vision for certain tasks is *5*" of an arc (*2.42* 10^{-4} radians), *5* times larger than the acuity of one cone. For our synthetic examples we made the screen (synthetic retina) have a size of 100 units distance and a density of 1 pixel each 2.5 units distance. The typical value of a focus is *50* units distance. This implies that each pixel is responsible for *2.8 degrees* (*0.05 radians*). If the value of the focus is increased the acuity

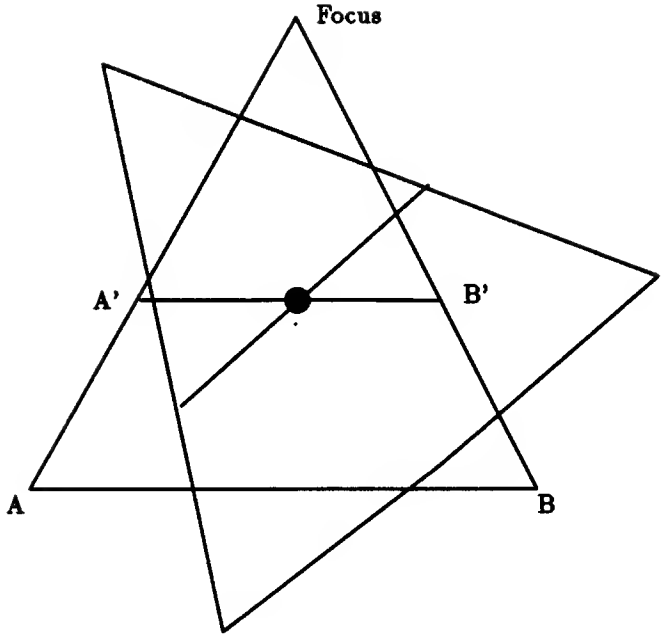

Figure 5: *The "virtual fovea" given by the line A'B' and the reduced focus defines the virtual eye. The fovea is defined by the line AB and capture the same elements in the scene as the virtual fovea. The virtual eye is described by the same model as the artificial eye, e.g. the center of rotation is located on the "virtual fovea".*

is also increased by the same factor. Increasing the number of pixels in the screen also increases the acuity by the same factor. The difference between these two different methods of increasing the acuity of the eye system is that for the first case (changing the focus) the visual field is also changed. More precisely, suppose we are looking at an object and we increase the focus, this corresponds to "zooming" towards this object. The zooming process gives us more acuity but also restricts us to a smaller visual field. In our examples, described in chapter 5, the number of pixels was kept constant and whenever we needed more accuracy the focus was increased. We have to point out that for a given camera the number of pixels is fixed and so there is no freedom to change it.

The human eye, unlike our model, does not rotate about the center of its image plane. As illustrated in figure 5 we can model the human eye by a virtual eye with reduced focus and increased density which does rotate about the image plane. The virtual eye has a focus of 10 mm, a width of 1.5 mm and a density of 700 cones per mm.

We must compare the distance measurements in our model to those of the human eye. In our model the distance between the centers of the two eyes was 100 units, for the human eye it is roughly 6 cm. This gives us a conversion rate: 1 unit = 0.06 cm. For example, the focus of our eye model, 50 units, corresponds to 3.0 cms. Similarly the density of receptors, 256 pixels per 100 units, corresponds to 4.2 per mm. The typical depths we considered was 1,200 units, or 72 cm.

To compare the parameters of our model to the human eye we must see how the errors scale with distance, lattice spacing, amount of eye-rotation and focal length. The human eye has considerably smaller lattice spacing. On the other hand its focal length is also smaller and its angle of rotation cannot be as large. We would also like the system to work for distances larger than 72 cms.

The amount of angle rotation is constrained because it is necessary to keep the object on the fovea, or macula lutea. Figure 6 shows the relation between the angle of rotation $\delta\omega$, the angle subtended by the image plane ω and the angle of view for which objects will be maintained on the fovea, ρ. The relationship is

$$\omega - \delta\omega = \rho. \tag{4.1}$$

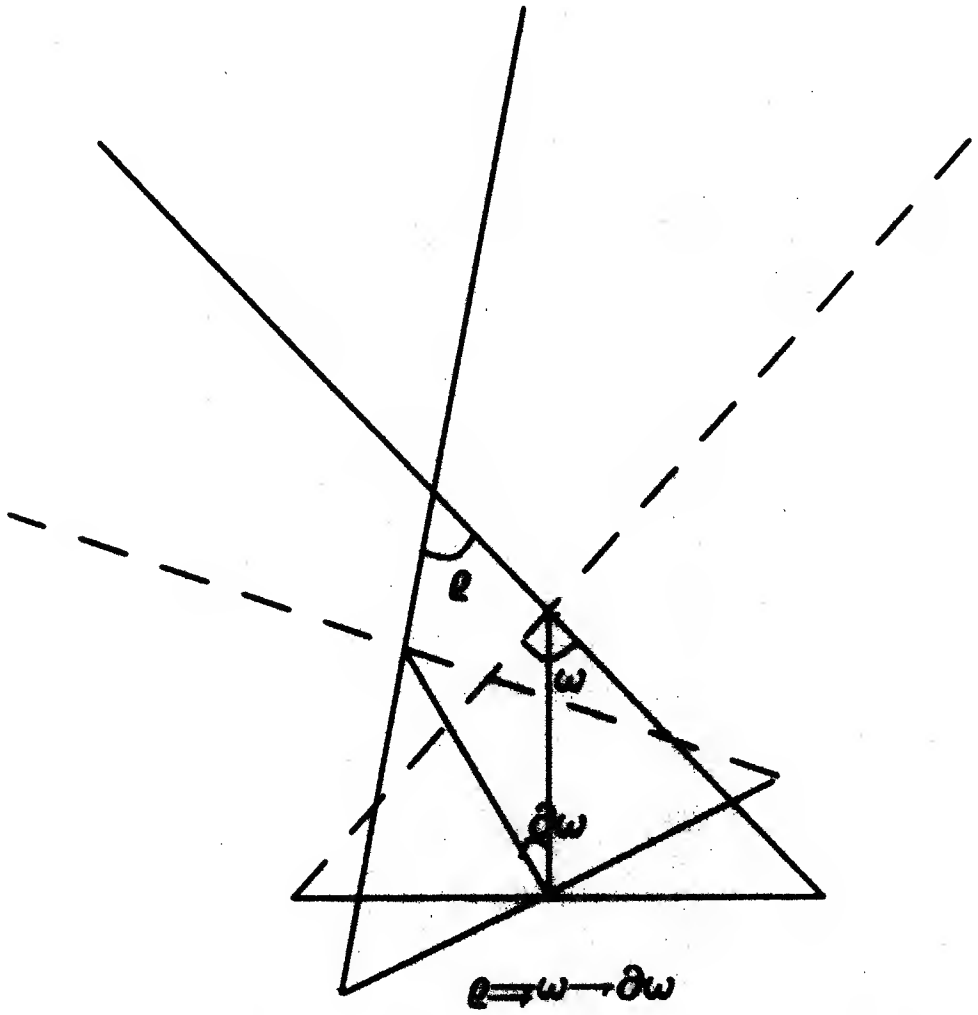

Figure 6: *Due to the rotation of the eye (δω) the viewing angle is reduced. Some features in the first frame are not seen in the second frame.*

For the human eye $\omega = 0.15$ radians, for our model it is 1.57 radians (or forty five degrees). Thus the angle of view for the human eye is very small.

We now see how the errors in estimating the depth from each eye scale. For simplicity we consider the projection of a point, $(x, y, 0)$, lying on the horizontal meridian. We assume the right eye is initially pointed directly forward and then rotates by an angle ϕ. The error in the depth estimate δy is given by (5.4b) in terms of the positions of the projected points on the images, x_{R1} and x_{R2}. To see how δy scales with depth we must substitute for x_{R1} and x_{R2} in terms of x and y. We use (3.11) and expand in inverse powers of y, this gives

$$x_{R1} = \frac{l}{y} + O\Big(\frac{1}{y^2}\Big),$$

$$x_{R2} = tan\phi + \frac{l}{y} + \frac{(f + lsin\phi)tan\phi)}{ycos\phi} + O\Big(\frac{1}{y^2}\Big).$$

Substituting into (5.2, 5.3) gives

$$A = \frac{f^3 tan\phi}{y},$$

$$C = f^3 tan\phi + \frac{f^4 sin\phi - lcos\phi + l}{ycos^2\phi},$$

$$\delta A = f\delta_1\Big(-\frac{1}{cos\phi} - \frac{lsin\phi}{y} - \frac{(f + lsin\phi)tan^2\phi}{y}\Big) + f\delta_2(cos\phi - \frac{lsin\phi}{y}),$$

$$\delta C = f^2\delta_1\Big(-\frac{1}{cos\phi} - \frac{lsin\phi}{y} - \frac{(f + lsin\phi)tan^2\phi}{y}\Big) + f^2\delta_2(1 - \frac{lsin\phi}{y}).$$

This gives

$$\frac{\delta y}{y} = \frac{y}{f^2 sin\phi}(-\delta_1 + \delta_2 cos^2\phi) + O(y^3).$$

Thus the error ϵ scales as

$$\epsilon = \frac{yL}{f^2 sin\phi}, \tag{4.2}$$

where L is the size of the lattice spacing.

Since the $f_A = 3f_{VE}$ (the focus of the artificial eye, 30 mm, is three times the focus of the virtual eye, 10 mm.), $L_A = 320L_{VE}$ (lattice space) and assuming the angle of rotation of the eye to be 0.16 rad (one third of the artificial eye) then equation 4.2 gives

$$\frac{\epsilon_A}{\epsilon_{VE}} = 11\frac{y_A}{y_{VE}} \tag{4.3}$$

For the case where the ratio $\frac{\epsilon_A}{\epsilon_{VE}} = 1$ and since in our test $y_A = 1200 units = 72cm$ it imply for humans a capacity of dealing with depth of 8 meters with the same performance as in our tests.

5 Error Analysis

5.1 Why and how to use Error Analysis

Suppose we have two corresponding points in the left eye at two different angles of rotation. Using the formulae in the previous chapter we will be able to identify the corresponding point in space, if we know the angle between the two frames. Then the stereo matching process would be solved (but redundant), since by projecting this point into the right eye we would find the corresponding stereo point. However, if there is noise in the system or any kind of error source, the estimate of this point may be poor, particularly for small angles. For our system the chief source of error is due to the lattice spacing. A point projected onto the image screen is assigned to the nearest lattice point, and thereby has a possible error of up to one half of the lattice spacing. For small angles this can give rise to enormous errors in the estimates of the points in 3-D, see figure 7.

In this section we show how we can derive a probability distribution for such errors. This distribution can then be used to test (*rotation depth test*) a possible match. For example, suppose a pair of points in the left image has estimated depth of *(23.0, 1200.3, 34.7)* and suppose a pair of points in the right image have an estimated depth of *(10.4, 1156.1, 13.0)*. We can calculate the probability that the left and the right points correspond (i.e. that the difference in their estimates is merely due to a lattice spacing error). If this probability is above a certain threshold (one of the control parameters) then

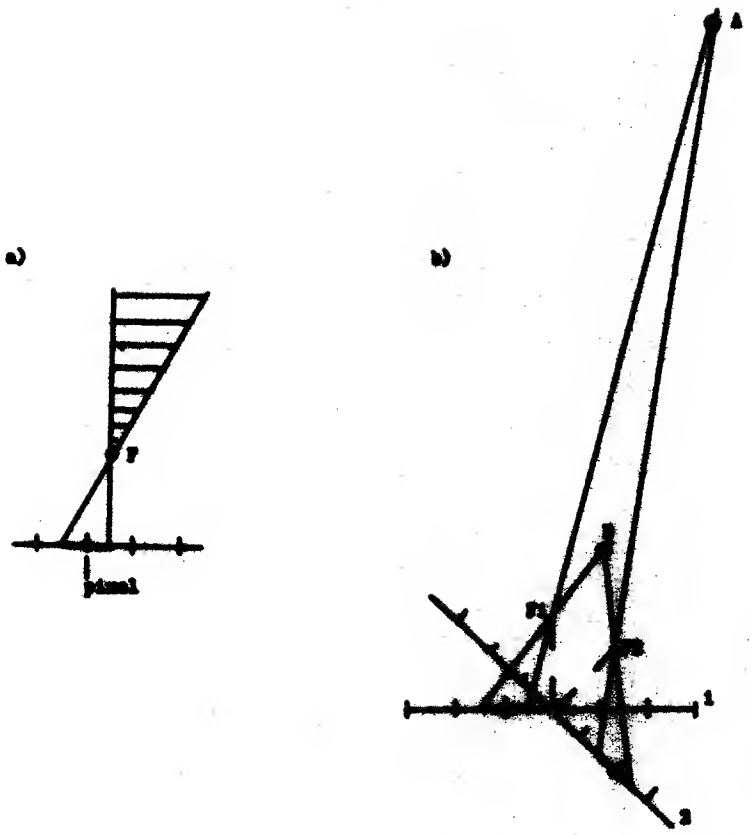

Figure 7: *a) The entire shaded region is projected to the same pixel due to the error from discrete lattice spacing. b) Points A and B, although very apart in 3D-space, are projected into the same pixel in both frames.*

the match passes the rotation depth test. This threshold is initially high, so that only points with close estimates pass the test, and is slowly lowered as points are matched and the matching ambiguity is reduced.

5.2 A probability distribution for the lattice errors

We have two frames for the right eye at angles Φ_1 and Φ_2 respectively. We find the correspondence between these two frames (by a method described in the next section). If a point $p_{R1} = (x_{R1}, z_{R1})$ in the first frame corresponds to a point $p_{R2} = (x_{R2}, z_{R2})$ in the second frame we will refer to the pair p_{R1}, p_{R2} as P_R. The left eye also has two frames at angles Ψ_1 and Ψ_2. The following derivations will be for the right eye only. The results for the left eye can be found by replacing the Φ's with Ψ's and sending $l \mapsto -l$.
We now consider the right eye. Suppose we have two corresponding points (x_{R1}, z_{R1}) and (x_{R2}, z_{R2}) in frames with angles Φ_1 and Φ_2 respectively. These correspond to a point (x, y, z) in 3D-space with

$$x - l = \frac{B}{A} \qquad y = \frac{C}{A} \qquad z = -\frac{z_{R1}}{f} D_1, \tag{5.1}$$

where

$$A = (x_{R1}x_{R2} + f^2)sin(\Phi_1 - \Phi_2) + f(x_{R2} - x_{R1})cos(\Phi_1 - \Phi_2), \tag{5.2a}$$

$$B = fx_{R1}x_{R2}(cos\Phi_2 - cos\Phi_1) - f^2 x_{R1} sin\Phi_2 + f^2 x_{R2} sin\Phi_1, \tag{5.2b}$$

$$C = fx_{R1}x_{R2}(sin\Phi_1 - sin\Phi_2) - f^2 x_{R1} cos\Phi_2 + f^2 x_{R2} cos\Phi_1, \tag{5.2c}$$

$$D_1 = \frac{B}{A} sin\Phi_1 + \frac{C}{A} cos\Phi_1 - f, \tag{5.2d}$$

The errors arise from the lattice spacing errors of $x_{R1}, x_{R2}, z_{R1}, z_{R2}$. Let the errors in x_{R1}, x_{R2} to be δ_1, δ_2 and the errors in z_{R1}, z_{R2} to be $\delta z_{R1}, \delta z_{R2}$. In the first order of δ_1, δ_2 the errors in A, B and C are obtained (the higher

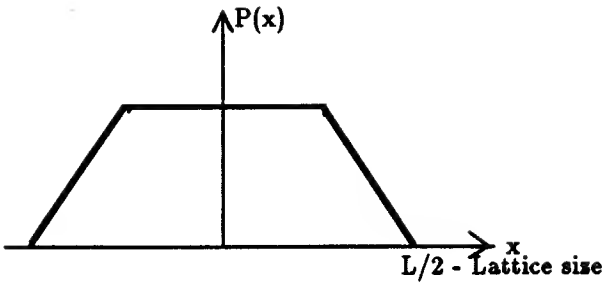

Figure 8: *Probability distribution for the error due to the lattice spacing*

order error terms are always smaller and can be neglected). For example, the error in A then is

$$\delta A = (\delta_2 x_{R1} + \delta_1 x_{R2}) sin(\Phi_1 - \Phi_2) + f(\delta_2 - \delta_1) cos(\Phi_1 - \Phi_2). \qquad (5.3)$$

Since $\delta x = \frac{\delta B}{A} - \frac{B\delta A}{A^2}$ and so on for $\frac{\delta y}{y}$, δz, we can then write

$$\delta x(\delta_1, \delta_2) = \alpha_1 \delta_1 + \alpha_2 \delta_2, \qquad (5.4a)$$

$$\frac{\delta y}{y}(\delta_1, \delta_2) = \beta_1 \delta_1 + \beta_2 \delta_2, \qquad (5.4b)$$

$$\delta z(\delta_1, \delta_2, \delta z_{R1}) = \gamma_1 \delta_1 + \gamma_2 \delta_2 + \gamma_3 \delta z_{R1}, \qquad (5.4c)$$

where the α, β, γ are determined from (5.1) and (5.3). It is important to note that their values depend on the position of the lattice and are calculated separately at each lattice point.
The δ's are assumed to be distributed independently in the range $-L/2, +L/2$, where L is the lattice spacing. The formulae (5.4) therefore define a probability distribution for the errors. These distributions have the shape shown in figure 8. The mean of these distributions is zero (by symmetry) and it is straightforward to calculate their standard deviations. We denote these by $\sigma_x, \sigma_y, \sigma_z$.

5.3 How to use the computed Error

Test 1: rotation depth test
We can now formally define the *rotation depth test.* Let P_R and P_L be points in the right and left eyes which are possible matches. They estimate points (x_R, y_R, z_R) and (x_L, y_L, z_L) in space and have standard deviations $\sigma_{x_R}, \sigma_{y_R}, \sigma_{z_R}$ and $\sigma_{x_L}, \sigma_{y_L}, \sigma_{z_L}$ respectively. There are three control parameters C_1, C_2, C_3 corresponding to each of the σ's (in practice these control parameters are usually chosen to be equal). P_R and P_L will pass the test provided the following are all satisfied

$$|x_L - x_R| \leq C_1(\sigma_{x_R} + \sigma_{x_L}), \quad |y_L - y_R| \leq C_2(\sigma_{y_R} + \sigma_{y_L}), \quad |z_{1L} - z_{1R}| \leq C_3(\sigma_{z_{1R}} + \sigma_{z_{1L}}). \tag{5.6}$$

Test 2: ratio test
While testing these results we discovered another regularity. For a large number of points the vector α_1, α_2 is almost parallel to β_1, β_2. This means that

$$\delta x = ratio(\alpha, \beta)\frac{\delta y}{y}, \tag{5.6}$$

where $ratio(\alpha, \beta)$ is the ratio of the lengths of the vectors α_1, α_2 and β_1, β_2. In other words it means that the error in the x estimate is a known multiple of the error in the y estimate. We use this regularity to define the *ratio test.* This is controlled by a parameter C_4. Figure 9 illustrates the *ratio test.*

Test 3: stereo test
We now define a final test for consistency, the *stereo test.* For potential matches p_R and p_L in the first frames of the two eyes we calculate the point $P - 3d$ in 3D-space which gave rise to them (note that such a point only exists if p_R and p_L lie on corresponding epipolar lines). We now project $P - 3d$ onto the second frames, as shown in figure 2. We calculate the distance in terms of lattice spacing between these projected points and the nearest points in the lattice. If these distances are smaller than the control parameters C_5 and C_6 then the points pass the stereo test. Initially C_5 and C_6 are zero, i.e. we require that there are points exactly where the projection exists.
We suggest that the error analysis described here is applicable to many other problems.

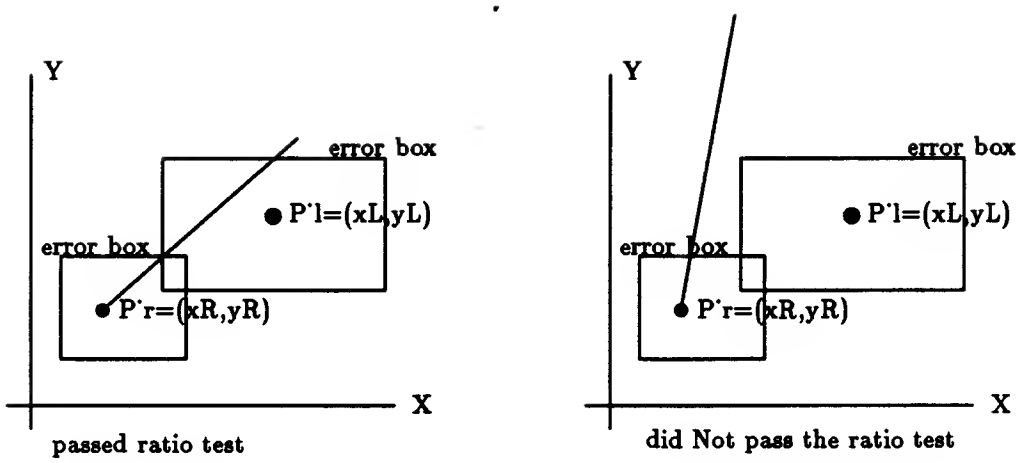

Figure 9: *The ratio test. For candidate matches P_R and P_L we test each to see if the ratio test is applicable (i.e. for each point we see if the α's and β's are almost parallel). If it applies to P_R we draw a line in the x, y plane passing through (x_R, y_R) and with the tangent given by the ratio of α_1 and β_1. Then we see how close this line lies to the box centred on (x_L, y_L) with sides of length proportional to $\sigma_{x_L}, \sigma_{y_L}$. If the line intersects the box, or passes by less than the control parameter C_4 from it then P_R passes the test. We then test P_L similarly, if the test is applicable.*

6 The Matching Process

We now describe the control strategy of the matching process. The first stage is to consider the right and left eyes separately and perform the correspondence as the eye rotates. We will describe this for the right eye only.

6.1 The right eye match

We have two frames at angles Φ_1 and Φ_2. We then inserted a number of frames (usually 3) with angles between Φ_1 and Φ_2. The forbidden zone (the region in which the ordering constraint is violated) is smaller for eye-rotations than for stereo, see figure 10. This allowed us to track points between frames. The matching was non-trivial, typically points moved between 13 and 21 lattice spaces from frame to frame. For each consecutive frames we first took the average motion of all the points and used this as an initial estimate for the match for each point. For each point we had an estimate, obtained from matching previous frames, of whether the point was moving faster or slower than the average. We used this estimate, and the new average motion, to be the centre of a small region (typical size was 4 pixels) in which we looked for a match. This computation was done scanning from left to right so that the ordering constraint was implicitly used, points already matched were removed and hence could only be matched once. For the choice of angles we tried 5 frames seemed optimal. Occasional mistakes were still made, but for our examples the match rate was about ninety-five percent. It would be interesting to compare this matching strategy with a cooperative, or energy function minimizing strategy, such as described in Ullman (1979)[17] and Gryzwacz and Yuille (1986)[4].

Once the rotation matching has been done in the two eyes separately we can calculate the estimated depths and errors. For each corresponding pair p_1 and p_2 we calculate the point P in 3D-space which projects to them. We also calculate the σ's for each pair. Finally we project P into the other eye to find the *corresponding point*, this is used as an initial guess for the match.

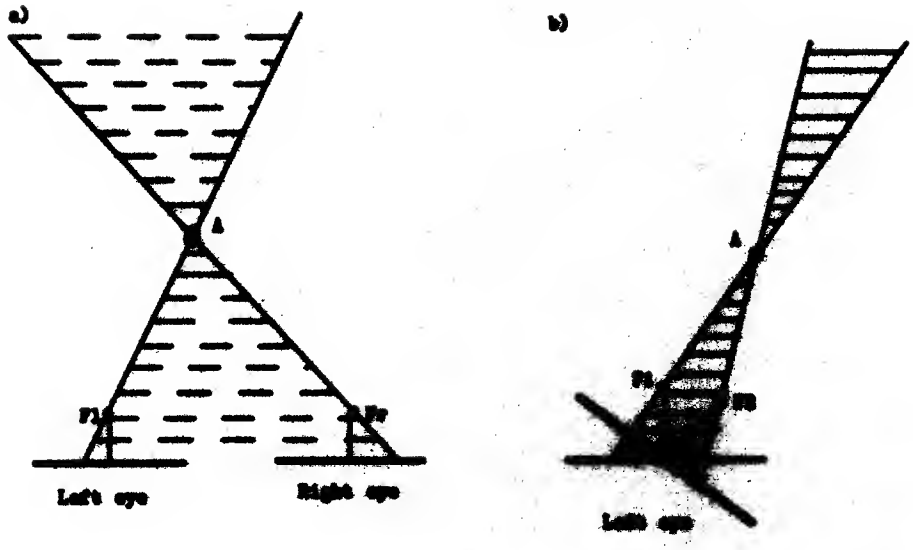

Figure 10: *The forbidden zone for a) stereo, larger base line. b) eye rotation, smaller base line.*

7 The stereo match

Now we proceed to the stereo matching. The control parameters $[C_i]$ defined in the previous section are set to their initial values. We define three new control parameters C_7, C_8 and C_9. The first two control the range, in the x and z directions, in which we look for possible matches. C_9 allows us to violate the epipolar constraint, which is sometimes necessary because of lattice discretization errors. For each point in the right eye we find the corresponding location in the left eye and search for possible matches in a region about this defined by the range parameters C_7, C_8. If possible matches exist we check to see if they pass the rotation test (controlled by C_1, C_2, C_3), the ratio tests (controlled by C_4) and the stereo test (controlled by C_5, C_6). If they pass all these tests they are matched and their points removed from the four arrays (two for each eye), so as not to confuse other matches. We loop over all points in the left eye removing points when they are matched. Then the algorithm automatically alters the control parameters and the process is repeated. The way we alter the control parameters, or the relaxation of the parameters, is clearly critical for this process. The strategy for the relaxation is highly conservative. We typically have between twenty and thirty loops. Figure 11 shows the role of all control parameters.

An optional feature of the algorithm is the *zooming* technique. This allows us to automatically examine regions where there are many points, and hence many potential errors. Once the region has been determined, for example if the density of points exceeded a threshold, the zooming technique rotated the eyes to verge towards the region and calculated the largest focal length for which the region still lay on the image screen. This increased the resolution and made the program more accurate.

8 Examples and Results

We tested our algorithm on a number of different types of stimuli. We now describe the performance of the algorithm on typical examples of these stimuli. When not specified the default value of the focus is 50. The angles for the eyes are *0.1,0.5 radians* for the left eye and *-0.1,-0.5 radians* for the right eye. These correspond to angles of $\pm$ *5.6* and $\pm$ *28.0 degrees*. The effective length of the lattice is 99% for the first angle and 88% for the second angle.

CONTROL PARAMETERS	ROLE OF THE PARAMETER
C1, C2, C3	ROTATION TEST (size of the box error)
C4	RATIO TEST (correlation between error in x and y)
C5, C6	STEREO TEST (check the projection of candidate in other multiple frames)
C7, C8	Size of the region where to search for matches (for x and z coordinates in the lattice)
C9	Violates the epipolar constraint

Figure 11: *The role of the Control Parameters*

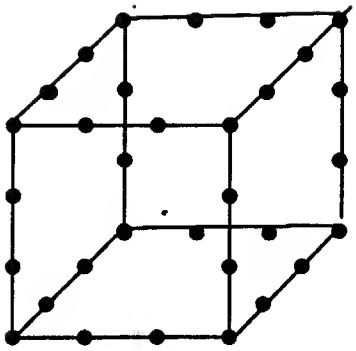

Figure 12: *The rectangloid*

The first stimuli is a rectangloid, shown in figure 12. It is a wire figure with feature points marked at regular intervals on the boundary. It is transparent, so the depth values are not continuous.

The second example is the double nail illusion, see figure 1. Not surprisingly our program avoids the illusion unless the nails are very close together. Thus it performs better than humans on this stimuli.

For the third example we investigate the occluding boundary of a cylinder, figure 13. As this boundary is smooth the boundary point seen by one eye will not correspond to the boundary point seen by the other eye. We investigate to see whether our program will match in this case.

Finally we consider a transparent random dot stereogram. This is shown in figure 14.

8.1 Rectangloid

For the first example, see figure 12, the rectangloid has dimensions $320 \times 480 \times 640$ pixels. The rectangloid is made up of wires and it is transparent. The ori-

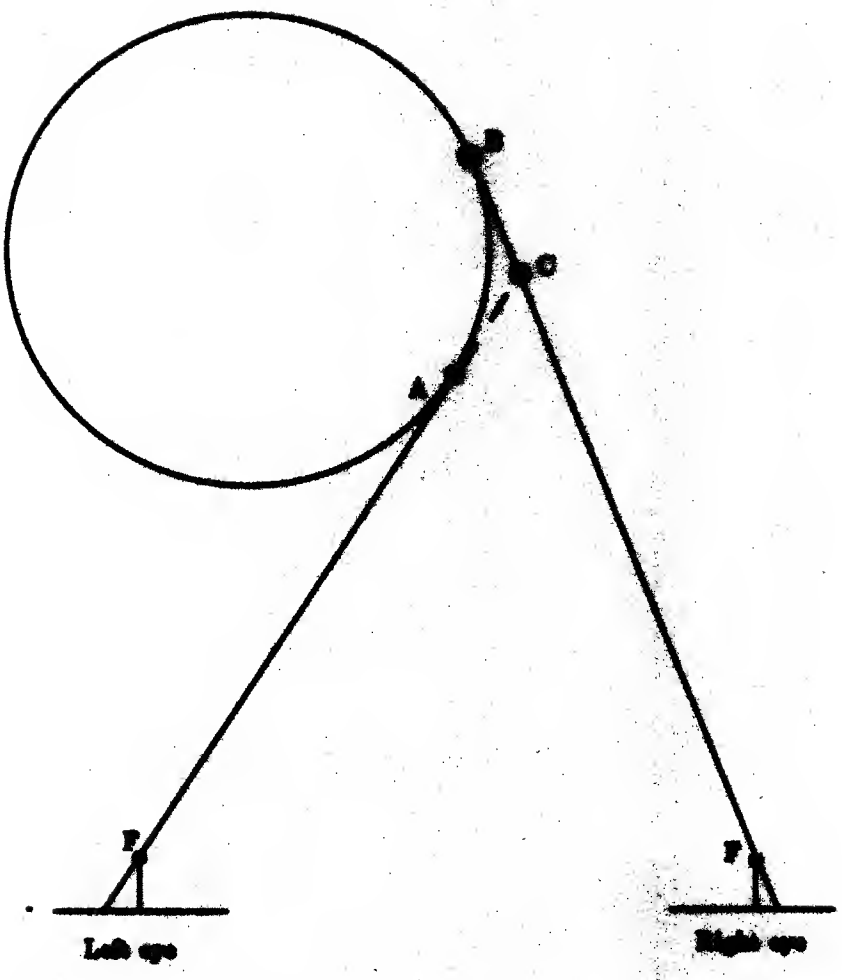

Figure 13: *Two dimensional slice of a cylinder and the occluding boundary.*

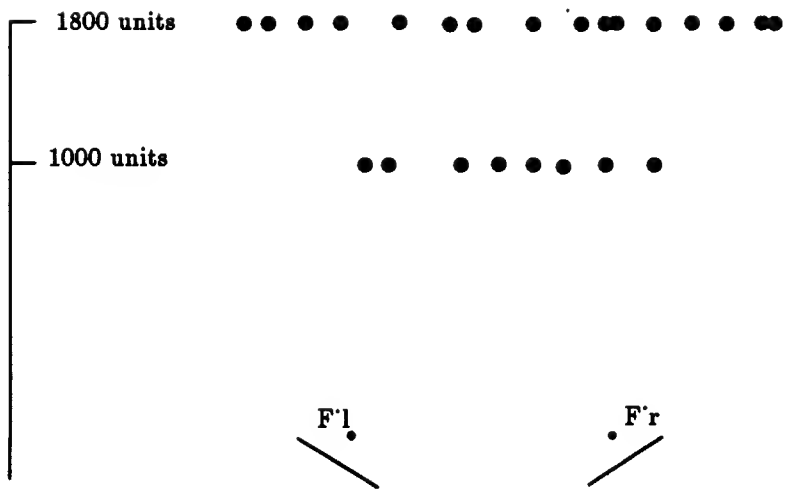

Figure 14: *Transparent random dot stereogram*

entation of the rectangloid is defined by the vectors $(1,0,0)$ $(0,1,0)$ $(0,0,1)$ with the origin vertex at *(0.0 1200.0 0.0)* . The density of dots is defined by setting a point every 160 distance units, with a total of 32 points. We rotated the eyes from an angle of *0.1 rad* to *0.5 rad* in 5 frames. The program got two errors. These two errors are the type of error produced in the double nail illusion discussed below. The remaining 30 points were corrected matched.

In the second example we choose a cube with size 320 X 320 X 320 pixels. The orientation of the cube is defined by the vectors $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ $(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ $(0,0,1)$ with the origin vertex at *(0.0, 1200.0, -100.0)* . The density was reduced to a point every 80 pixels, with a total of 44 points. We rotated the eyes from an angle of *0.1 rad* to *0.5 rad* in 5 frames. The program just got one error. Three point were unmatched but the default values obtained from the recovery of the left or right eyes movement is very good. The remaining 40 points were corrected matched.

We repeated the experiment with the same rectangloid (cube) keeping the same parameters but with a smaller number of frames, 3 frames. The program got 11 wrong matches. We conclude that multiple frames are necessary in order to trace correctly the points with eye rotation.

8.2 Double Nail illusion

Assume two point in space are fixed with the same coordinates x and z but slightly different depth y. See figure 1. They project into two points in each eye. Since the ordering constraint is violated , an algorithm based on ordering constraint gives the wrong matches. The human system often gets the wrong matches. Our algorithm, though, is very robust for correctly solving the double nail illusion.

For the first example we choose the data to be the points *(0.0, 1200.0, 0.0)* and *(0.0, 1230.0, 0.0)* . The algorithm correctly solved it. So a change in depth of 3% was correctly captured. Actually in this case even for changes in depth of 2% the algorithm gives the correct matches.

For the second example we choose points in the right side *(320.0, 1600, 320)* and *(320.0, 1680, 320)*. In order to get the correct matches, we made use of the zooming technique to find the best focus for projecting a given region. To avoid failing for the double nail illusion, the focus was automatically increased to 60.

In the rectangloid example we had two errors of the double nail type. Using the flexibility of the algorithm to change the focus we show that the mismatching can be corrected by increasing the focus, again using the zooming technique. For the points *(320, 1520, 0)* and *(320, 1680, 0)* and focus 60 the points are corrected matched.

8.3 Occluding boundary

We now consider occluding boundaries, where the boundaries of an object seen from the left and right eye are different. The example we use is a circle (2-dimensional) and is shown in figure 13. We choose the radius to be 100 and the centre to be at *(0,500,0)*. We find that a boundary in the left eye will never match the boundary in the right eye using our algorithm. So the estimated position of the boundary would come from retrieve-3d-rotation. A similar result was found for a circle with radius 200 and centre at *(0,1000,0)*. We conclude that this algorithm presents a strategy to deal with the problem of occluding boundaries, since it will not make false matches at the boundaries and will return a depth value for the boundaries based on eye rotation. It should be possible to develop this result into an occluding boundary detector; if two neighboring points were found with similar depth values (estimated

from retrieve-3d-rotation) but neither of which were matched then we would suspect an occluding boundary. This would be strengthened if stereo matching the two points gave a point whose depth lay between the depths of the points estimated by retrieve-3d-rotation, see figure 13.

8.4 Random dot stereogram

The final example is a random dot stereogram. We generate two squares of random dots, one at depth of 1000 and size 250 and the other at 1800 with size 400. For each square we generated 20 dots. This random dot stereogram differs from the standard type (Julesz 71[8]) because it is transparent and three dimensional. The results confirm the robustness of the algorithm. Three points had the wrong matches, but they were points very close to each other so the retrieved depth was practically identical to the true depth.

9 Extensions

A problem in all stereo algorithm is the registration [14] of the image. Following a suggestion by T. Poggio this algorithm could be used to do the registration. The tracking could be done in each eye and conceivably the registration would be derived from the tracking and depth estimation. However more investigation is necessary.

We could make more use of the zooming device, allowing the eyes to both rotate and change focus and zooming in to certain regions where the matching was ambiguous. For example, we could use this device to segment the image by finding occluding boundaries and then use it as an adjunct to a stereo algorithm using the ordering constraint.

The zooming device fits naturally into our strategy of doing the most likely matches first; we would only do matches that passed a certain probability threshold and then zoom in to do the remaining matches. Instead of rotating the eyes keeping the focus fixed we could have kept the eyes fixed and varied the focus [2]. It would be simple to modify our programs to do this and we plan to test it.

Systems using eye-rotations, or two cameras rotating, have inherent limitations because of the limited size of such rotations and the finite size of

[2]Following a suggestion by T.Binford.

the lattice (and errors in the system). Our method could be generalized to more general types of motion, such as translation of the eye (camera) system. Waxman and Duncan [18] describe using head rotation to help stereo matching. Our scheme can easily be adapted to such motions.

A final extension is to the combination of stereo and motion. There has been some work in this area recently [18][7]. Our method is not directly applicable since for eye movements (and for head rotation) the amount of eye movement (and head rotation) is known. We will not, however, be able to use motion directly to give a depth estimate. However it should be possible to modify our strategy. If two points are possible matches in 3-space their projected motions in the two eyes will be related. Thus we can define a *motion test* for stereo matching which only passes points which have consistent motions in the two eyes. The error analysis of section could be adapted to give probability distributions for such motions. In such a scheme motion would be chiefly used to disambiguate possible stereo matches. The recovery of structure would be mainly left to stereo. This is being investigated.

All the work on eye-motion is inherently parallelizable and can (we hope) be speeded up to work in real time. The matching between different rotation frames implicitly used a form of the ordering constraint (by scanning from right to left), but in practice this rarely seemed important for the matching. Another possibility is to replace our rotation matching scheme with a minimal matching scheme (Ullman 1979, [4]). We are implementing the algorithm to run on the Connection Machine (a parallel computer with $16K$ processors) at MIT. It can then be connected to the head-eye system built at the M.I.T. A.I. Lab consisting of two cameras capable of rotation. Theses cameras will have zoom capability and high precision for the rotation angles. This head-eye system is described in Cornog (1985)[2].

Finally we must extend this system to deal with real images (produced by the head-eye system). To do this we must extract features such as edges from the image and use them as the matching primitives. Two main modifications will be necessary:

(i) We will have to modify our error analysis to deal with the errors introduced in the edge detecting process. As described earlier we will need to have some estimate, found by local computation, of the possible error in localization of the edge. We suggested that the measure of localization defined by Canny (1985)[1] to define his edge detector could be used as a local measure of such an error. There will also be a second source of error if

the edge lies on a surface at a large angle to the viewer. This arises if the edge detector operates over a finite extent and can be analyzed. An alternative method would be to use the eye rotation itself to help detect the edges.

(ii) Matching lines rather than dots introduces the aperture problem [10]. For the matching in the left and right eyes this can be dealt with by requiring smoothness of the motion field [5]. For the stereo matching it will correspond to a form of figural continuity constraint. The epipolar line constraint will be enough to avoid the aperture problem, but figural continuity will be a useful addition.

10 Conclusion

We have described an algorithm for stereo with eye movements and demonstrated it on a number of different examples of dot figures. Unlike many stereo algorithms, it does not use an explicit (or implicit) assumption of smoothness for the viewed surface and can therefore deal with transparent surfaces. It is capable of detecting whether a boundary is occluding and is not easily fooled by the double nail illusion.
We have suggested that for humans eye movements may play a more important role than is currently attached to it.
The algorithm works by using an error analysis to give a probability distribution, based on the matching of points between different rotated frames, for the position of the point in space. This probability distribution is used to test for matches between points in the left and right eyes. The test is initially severe, so that only the most likely matches are made. It is then systematically relaxed as the increasing number of matches reduces the possible ambiguity. We suggest that this strategy can also be used on other problems. We specifically consider head-eye movement and stereo and motion. We discuss extensions to the basic algorithm such as zooming.

Acknowledgments We thank T. Poggio, B. Horn, A. Verri and J. Little for helpful comments.

References

[1] John F. Canny. *Finding Lines and Edges in Images.* Technical Re-

port TM-720, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1983.

[2] K. H. Cornog. *Smooth Pursuit and Fixation for Robot Vision.* Master's thesis, Massachusetts Institute of Technology, 1985.

[3] J. D.Krol and W. A. Van der Grind . The double-nail illusion: experiments on binocular vision with nails, needles and pins. *Perception*, 11:615–619, 1982.

[4] N. M. Gryzwacz and A. L. Yuille. *Motion Correspondence and Analog Network.* A.I. Memo No. 888, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1986.

[5] Ellen C. Hildreth. *The Measurement of Visual Motion.* MIT Press, Cambridge, Mass., 1984.

[6] J. Mayhew J. Pollard and J. Frisby. *Disparity Gradients and Stereo Correspondences.* Preprint, Sheffield Univ. Dept. of Psychology, 1984.

[7] M. R. M. Jenkins. *The Stereopsis of time-Varying Images.* Technical report in Biological and Computational Vision, University of Toronto, 1984.

[8] B. Julesz. *Foundaton of Cyclopean Perception.* University of Chicago Press, 1971.

[9] H. C. Longuet-Higgins. The role of the vertical dimension in stereoscopic vision. *Perception*, 11:377–386, 1982.

[10] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proc. R. Scoc. London B*, 211:151–180, 1979.

[11] David Marr and Tomaso Poggio. A theory of human stereo vision. *Proc. R. Scoc. London B*, 204:301–328, 1979.

[12] J. E. W. Mayhew. *Physical and Biological Processing of Images*, page . Springer-Verlag, Berlin, 1982.

[13] V. J. Milenkovic and T. Kanade. Trinocular vision using photometric and edge orientation constraints. In *Proceedings of the Image Understanding Workshop*, pages 163–175, Los Angeles, CA, December 1986.

[14] K. Nielsen and T. Poggio. Vertical image registration in stereopsis. *Vision Res.*, 24:1133–1140, 1984.

[15] G. F. Poggio and T. Poggio. The analysis of stereopsis. *Ann. Rev. Neurosci.*, 1984.

[16] K. Prazdny. *Detection of Binocular Disparities.* Preprint, 1984.

[17] Shimon Ullman. *The Interpretation of Visual Motion.* M.I.T. Press, Cambridge, MA, 1979.

[18] A. M. Waxman and J. H. Duncan. *Binocular image flows: steps toward stereo-motion fusion.* preprint, University of Maryland, 1985.

[19] A. L. Yarbus. *Eye Movements and Vision.* Plenum Press, 1967.

This blank page was inserted to preserve pagination.

CS-TR Scanning Project
Document Control Form

Date : 10 / 18 / 95

Report # AIM-927

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 35(40 - IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☒ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number):_____

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description :	Page Number:
IMAGE MAP: (1-35) UNNUMBERED TITLE PAGE,	1-34
(36-40) SCAN CONTROL, DOD, TRGT'S (3)	

Scanning Agent Signoff:
Date Received: 10 / 18 / 95 Date Scanned: 10 / 24 / 95 Date Returned: 10 / 25 / 95

Scanning Agent Signature: Michael W. Cook

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AI Memo 927	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: AD-A193588	
4. TITLE (and Subtitle): Stereo and Eye Movement	5. TYPE OF REPORT & PERIOD COVERED: memorandum
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Davi Geiger and Alan Yuille	8. CONTRACT OR GRANT NUMBER(s): N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: January 1988
	13. NUMBER OF PAGES: 34
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, VA 22217	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

stereo
error analysis
eye movement
controlled movement

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We describe a method to dolve the stereo correspondence using controlled eye (or camera) movements. These eye-movements essentially supply additional image-frames which can be used to constrain the stereo matching. Because the eye-movements are small, traditional methods of stereo with multiple frame will not work. We develop an alternative approach using a systematic analysis to define a probability distribution for the errors. Our matching strategy then matches the most probable points first, thereby reducing the ambiguity for the remaining matches. We demonstrate this algorithm with several examples.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Completed 7-28-88

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**